Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.2, Apr. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.2.391

Gohr의 Speck32/64 신경망 구분자에 대한 분석과 Simon32/64에의 응용*

성 효 은,[1†] 유 현 도,[1] 염 용 진,[2‡] 강 주 성[2]
[1]국민대학교 금융정보보안학과 (대학원생), [2]국민대학교 정보보안암호수학과 (교수)

Analysis of Gohr's Neural Distinguisher on Speck32/64 and its Application to Simon32/64*

Hyoeun Seong,[1†] Hyeondo Yoo,[1] Yongjin Yeom,[2‡] Ju-Sung Kang[2]
[1]Dept. of Financial Information Security, Kookmin University (Graduate student),
[2]Dept. of Information Security, Cryptology and Mathematics, Kookmin University (Professor)

요 약

Aron Gohr는 경량 블록암호 Speck에 대해 딥러닝 기술에 기반한 암호분석 기법을 제안하였다. 이는 기존의 차분분석 방식보다 높은 정확도로 선택적 평문 공격을 가능하게 한 방법이다. 본 논문에서는 이러한 딥러닝 기반 암호분석의 동작 원리에 대해 확률분포를 이용하여 분석하고 이를 경량 블록암호 Simon에 적용한 결과를 제시한다. 또한, 암호분석 알고리즘 내부에서 신경망의 예측값 확률분포가 Speck과 Simon의 각 라운드 함수 특성에 따라 차이가 있음을 규명한다. 이를 통해 Aron Gohr가 제시한 암호분석의 핵심기술인 신경망 구분자의 성능 개선 방향을 제시한다.

ABSTRACT

Aron Gohr proposed a cryptanalysis method based on deep learning technology for the lightweight block cipher Speck. This is a method that enables a chosen plaintext attack with higher accuracy than the classical differential cryptanalysis. In this paper, by using the probability distribution, we analyze the mechanism of such deep learning based cryptanalysis and propose the results applied to the lightweight block cipher Simon. In addition, we examine that the probability distributions of the predicted values of the neural networks within the cryptanalysis working processes are different depending upon the characteristics of round functions of Speck and Simon, and suggest a direction to improve the efficiency of the neural distinguisher which is the core technology of Aron Gohr's cryptanalysis.

Keywords: Differential cryptanalysis, Deep learning, Block cipher, Neural distinguisher

I. 서 론

1995년 J.McCarthy에 의해 용어가 정의된 인공지능(artificial intelligence)은 인간의 지성을 요구하는 각 분야의 문제를 컴퓨터 소프트웨어의 알고리즘을 활용하여 해결하는 기술을 의미한다. 이러

Received(02. 04. 2022), Accepted(03. 03. 2022)
* 본 연구는 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2021-0-00046, 국가공공 정보시스템 안전성 및 활용성 제고를 위한 차세대 암호체계 개발)
† 주저자, she000@kookmin.ac.kr
‡ 교신저자, salt@kookmin.ac.kr(Corresponding author)

한 인공지능 기술은 컴퓨팅 기술의 발달과 딥러닝(deep learning) 기술의 등장으로 인해 오랜 침체기를 벗어나 전성기를 맞고 있다. 암호 안전성 분석 분야에도 딥러닝 기술을 적용하려는 노력은 지속해서 있었다[1][2]. 그러나 이는 대부분 고전 암호에 국한된 결과이거나 딥러닝 기술의 핵심 요소인 신경망(neural network)의 이산함수 근사 이론[3]에 부합하는 신경망의 크기를 고려하지 않은 결과로 보인다. 실제로 [1]에서는 Cascade 구조의 신경망을 학습시켜 암호키 없이 DES 복호화에 성공하였다는 결과가 제시되었다. 그러나 이는 실험에 사용된 신경망의 복잡도가 매우 낮아 함수의 표현에 필요한 논리회로 개수의 하한을 제시하는 Shannon의 정리에 부합하지 않으며 실험의 기반이 되었을 것으로 보이는 [3]의 내용 역시 DES와 같은 이산함수에 대한 근사 가능성을 다루지 않는다는 점을 들어 반박되었다[4].

반면 지난 Crypto 2019에서 A. Gohr에 의해 발표된 논문은 딥러닝 기술을 경량 블록암호 Speck의 안전성 분석에 적용한 사례로 기존에 존재하던 차분분석 방식에서 구분자(distinguisher)의 성능을 딥러닝 기술을 사용하여 향상시킨 결과를 제시하였다[5]. 해당 연구를 통해 딥러닝 기술을 차분분석과 같은 기존의 암호분석을 보완하는데 적용하는 방향이 유의미한 결과가 있음이 입증되어 암호학계에서도 딥러닝 기술을 암호분석에 적용하는 것에 본격적으로 관심을 두게 된 것으로 보인다. 그러나 아직은 A. Gohr가 구분자의 성능향상을 위해 제시한 알고리즘 자체에 대한 학술적 분석은 미미한 수준이다.

본 논문에서는 딥러닝 기술을 적용한 암호 안전성 분석의 일환으로 A. Gohr가 제시한 구분자와 알고리즘을 심도 있게 분석하고, 이를 경량 블록암호 Simon에 적용한 암호분석 결과를 제시한다. 2장에서는 A. Gohr가 제시한 연구 성과를 비롯한 딥러닝 기반의 암호분석 동향을 소개하며, 경량 블록암호 Simon, Speck의 개요에 관해 설명한다. 3장과 4장에서는 Simon에 딥러닝 기반 구분자를 적용한 결과를 제시하고, 구분자의 성능향상을 위해 제시된 Key averaging 알고리즘의 원리에 대해 확률분포를 활용하여 분석한다. 마지막으로 5장에서 분석 내용을 토대로 구분자의 성능을 높일 수 있는 방향을 제시한다. 본 논문에서 제시하는 주요 성과를 정리하면 다음과 같다.

- A. Gohr가 제시한 딥러닝 기반 암호분석 기법을 Simon 암호의 구분자를 생성하는 데 적용한 암호분석 결과 제시.
- Key averaging 알고리즘이 구분자의 성능을 향상시키는 원인을 분석하기 위해 알고리즘의 입력 데이터에 따른 신경망 예측값의 확률분포를 관측하고, 라운드 함수에 따른 분포의 차이 규명.
- Key averaging 알고리즘의 내부 핵심 함수인 가중치 함수에 대한 실험적 분석을 통한 구분자의 성능향상 방안 도출.

II. 관련 연구

2.1 딥러닝 기반 암호분석 동향

딥러닝을 암호분석에 활용하려는 연구는 꾸준히 진행되고 있으며 공격의 종류는 크게 다음과 같이 분류할 수 있다[6].

- 키 복구 공격 (key recovery attack)
- 암호 근사 공격 (cipher emulation attack)
- 식별 공격 (identification attack)

그 중 [5]의 연구는 딥러닝 기술을 이용한 키 복구 공격의 일환으로써 경량 블록암호 Speck의 키 복구 공격에 사용되는 차분분석 방식을 신경망을 이용하여 보완한 방법을 제시하였다. 이후로 이러한 키 복구 공격에 딥러닝 기술을 적용한 후속연구들이 제시되었다. [7]에서는 DNN 모델을 S-DES, Simon, Speck 암호에 적용하여 분석한 결과를 제시한다. 해당 논문에서는 키 공간을 64개 이하의 ASCII 문자로 제한할 경우 Simon32/64와 Speck32/64의 전체 라운드에 대한 분석이 가능하다는 것을 제시한다. 이는 문자 기반(text-based)키로 키 공간이 제한되었다는 제약점이 있으나 각 암호 알고리즘에 대해 기존의 차분분석 방식보다 적은 데이터를 사용하여 높은 성공률로 키 복구를 가능하게 했다는 점에서 의의가 있다. [8]에서는 딥러닝 기술을 사용하는 경우 신경망의 입력 데이터에 대한 추가적인 분석을 진행하였다. 입력 데이터의 차분이 기존 차분분석 상에서는 동일한 차분확률을 가지더라도 활성화(activate)된 bit가 앞의 16-bit에 있는지, 뒤의 16-bit에 있는지에 따라 신경망의 구분 결과를 비교하였다.

[9]에서는 [10]에 제안된 차분 경로를 변형하고

[11]에 제안된 신경망 기반 구분자를 개선하여 이를 결합한 암호분석 방법을 제시하였다. 제시한 신경망 기반 구분자를 Present와 Simeck 암호에 적용하여 차분분석 기법을 사용한 기존 구분자보다 성능이 향상된 구분자를 제시하였다. [12]에서는 신경망 기반 구분자를 Simon과 Simeck 암호에 적용시켰다. 구분자의 성능을 향상시키기 위해 SAT/SMT Solver를 사용하여 효과적인 입력 차분을 발견하였으며 연관 키 차분 관점에서 신경망을 적용한 구분자를 제시하였다. 또한 A. Gohr의 연구에서 신경망의 입력으로 사용되는 데이터 형식을 개선하여 연관 키 차분 구분자의 성능을 향상시켰다.

2.2 딥러닝 기반 경량 블록암호 분석

[5]에서는 경량 블록암호 Speck32/64의 부분 라운드에서 기존 구분자보다 향상된 성능을 보이는 딥러닝 기반 신경망 구분자(neural distinguisher)가 제안되었다. 본 항에서는 경량 블록암호 Speck과 Simon의 동작 과정과 차분특성을 이용하는 구분자의 개요를 설명하고, [5]에 제안된 신경망 구분자를 파악한다.

2.2.1 Speck과 Simon

Speck과 Simon은 ARX 구조를 가지는 경량 블록암호로 2013년 NSA(National Security Agency)에 의해 공개되었다[13]. 각 블록암호는 n-bit의 워드 크기(word size)를 가질 때, Speck$2n/mn$, Simon$2n/mn$로 표기한다. 이때, $2n$은 블록 크기(block size), $k=mn$는 키 크기(key size)를 나타낸다. Speck과 Simon에 사용되는 연산은 다음과 같다.

- bitwise XOR, $\oplus$
- bitwise AND, $\wedge$, &
- j-bit left(or right) circular shift, $\ll$ j (or $\gg$ j)
- addition modulo 2^n, $\boxplus$

Speck32/64와 Simon32/64의 라운드 함수는 Fig.1.과 같이 유사 Feistel 구조를 가지며 입력, 출력 크기가 동일하다. 그 중 Speck32/64의 라운드 함수 구조가 상대적으로 더 복잡하며 확산(diffusion) 효과가 우수하다.

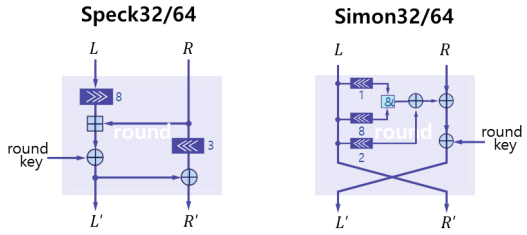

Fig. 1. Round function of Speck and Simon

- 입력: (L,R), $L,R\in GF(2)^{16}$
- 출력: (L',R'), $L',R'\in GF(2)^{16}$
- 라운드 키: $k\in GF(2)^{16}$

(1) Speck32/64

Speck32/64는 16-bit 워드 크기를 가지는 알고리즘으로 라운드 함수를 총 22번 반복한다. 라운드 함수는 식 (1)과 같다.

$$\begin{cases} L' = ((L \gg 8) \boxplus R) \oplus k \\ R' = (R \ll 3) \oplus ((L \gg 8) \boxplus R) \oplus k \end{cases} \quad (1)$$

이때, Speck32/64의 라운드 함수 차분확률은 $GF(2)^{16}$상에서의 덧셈 연산인 $\boxplus$ 연산에 의해 결정된다

(2) Simon32/64

Simon32/64는 16-bit 워드 크기를 가지는 알고리즘으로 라운드 함수를 총 32번 반복한다. 라운드 함수는 식 (2)와 같다.

$$\begin{cases} L' = (R \oplus ((L \ll 1) \wedge (L \ll 8)) \\ \qquad \oplus (L \ll 2)) \oplus k \\ R' = L \end{cases} \quad (2)$$

이때, Simon32/64의 라운드 함수 차분확률은 circular shift와 AND 연산이 결합된 $(L \ll 1) \wedge (L \ll 8)$ 연산에 의해 결정된다.

2.2.2 차분특성을 이용한 구분자

Biham과 Shamir에 의해 제안된 차분분석[14]은 블록 암호에 대한 키 복구 공격의 방법 중 하나로 랜덤함수와 블록암호의 구별되는 특징인 차분특성

(differential characteristics)을 이용하여 마지막 라운드 키를 복구하는 공격이다. 차분분석은 특정 차분을 가진 입력값으로부터 얻은 출력값의 차분을 이용하는 선택적 평문 공격(Chosen Plaintext Attack, CPA)이다.

S를 n-bit 입력값과 m-bit 출력값을 가지는 부울 함수(boolean function)라고 하자. 차분특성은 S의 차분이 랜덤함수와 구별되는 특성을 가지는 것을 의미한다.

$$S: GF(2)^n \rightarrow GF(2)^m,\ x \in GF(2)^n$$

S의 전체 가능한 입력쌍의 수는 2^n개이다. 입력차분이 a일 때 출력차분이 b인 입력쌍의 수 $\delta_s(a,b)$와 차분확률 $DP^S(a,b)$는 다음과 같이 표현할 수 있다.

$$\delta_S(a,b) = \#\{x \mid S(x) \oplus S(x \oplus a) = b\}.$$
$$DP^S(a,b) = \frac{\delta_S(a,b)}{2^n}.$$

최대 차분확률은 S의 가능한 모든 차분쌍 (a,b)에 대해 가장 높은 $\delta_s(a,b)$을 가질 때의 차분확률을 의미하며 다음과 같이 표기한다.

$$DP^S\max = \max_{a \neq 0, b} DP^S(a,b).$$

이때, S가 이상적인 랜덤함수라면 $DP^S\max = \frac{1}{2^m}$이 될 것이다. 그러나 부울함수 S는 일반적으로 $DP^S\max = \frac{1}{2^m} + \alpha\ (\alpha > 0)$값을 갖는다. 이러한 차분특성을 이용하여 S의 출력 결과와 랜덤한 출력 결과를 구분하는 것을 구분자라고 한다.

2.2.3 딥러닝 기반 Speck 차분분석

〔5〕에서는 기존 차분분석에서 사용하는 차분 구분자(Classical Differential Distinguisher, CDD)를 기반으로 딥러닝을 적용한 구분자를 제안하였다.

구분자 생성방식은 단일 신경망을 사용하는 방식과 신경망에 Key averaging 알고리즘을 적용하는 방식으로 볼 수 있다. 단일 신경망을 사용하는 방식은 신경망의 입력을 Speck32/64의 부분 라운드 암호화한 데이터 또는 랜덤 함수로 생성된 랜덤 데이터를 사용하여 두 데이터를 구분하도록 신경망을 학습한다. 즉 신경망이 구분자가 되며 이때 〔5〕에서 사용한 모델과 동일한 ResNet 구조의 CNN을 신경망 모델로 사용한다. Key averaging 알고리즘을 적용한 방식은 단일 신경망에 Key averaging 단계를 추가하여 구분자의 성능을 향상시키는 방식이다. A. Gohr가 제시한 Key averaging 알고리즘의 구체적인 과정은 Fig.2. 와 같다.

Algorithm 1 Key averaging algorithm

Input: Ciphertext pair $C_0, C_1 \in \{0,1\}^b$ (b: block size), r-round distinguisher Net
Output: (r+1)-round distinguisher output V $(0 < V < 1)$

1: **for all** $k \in$ subkeys
2: $\quad D_i[k] \leftarrow \mathrm{Dec1R}(C_i, k)$ //decrypt 1-round
3: $\quad z_k \leftarrow \mathrm{Net}(D_0[k], D_1[k])$
4: $\quad v_k \leftarrow z_k/(1-z_k)$
5: $v \leftarrow \mathrm{Average}([v_k, k \in \text{subkeys}])$
6: $V \leftarrow v/(1+v)$
7: **return** V

Fig. 2. Key averaging algorithm 〔5〕

Table 1.은 〔5〕에서 제시한 신경망 구분자의 정확도와 본 논문에서 〔5〕의 저자가 제시한 소스코드를 실험에 사용하여 Speck의 각 라운드에 신경망 구분자의 정확도를 측정한 결과를 나타낸다〔15〕. 두 실험은 거의 유사한 정확도를 보여주며 Gohr의 딥러닝 기반 신경망 구분자가 부분 라운드의 Speck과 랜덤한 출력을 유의미한 확률로 구분함을 뒷받침한다.

A. Gohr의 연구 결과를 정리하면 다음과 같다.

- Key averaging 알고리즘을 적용한 구분자가 기존 차분 구분자를 사용했을 때보다 더 높은 정확도를 가진다. 즉 랜덤한 함수로 생성된 랜덤 데

Table 1. Accuracy of distinguisher for Speck32/64

		CDD	Neural distinguisher [5]	Neural distinguisher 〔15〕
Round	6	0.758	$0.788 \pm 8.17\times10^{-4}$	0.796
	7	0.591	$0.616 \pm 9.7\times10^{-4}$	0.633

이터와 Speck32/64의 부분 라운드로 암호화된 암호문 데이터를 잘 구별한다.

- 9-라운드 Speck32/64에 대해 오직 128 개의 선택된 평문을 이용하여 라운드 키의 부분을 복호화하였으며 이 결과는 기존 차분 구분자를 사용한 것과 비교하여 라운드 키의 후보를 1/5로 줄인 것이다.

III. 딥러닝 기반 Simon 차분분석

A. Gohr가 제시한 딥러닝 기반의 암호분석은 기존의 차분분석 방식에서 암호문에 나타나는 차분특성(differential characteristic)을 신경망을 이용해 학습한 것이다. 본 장에서는 [5]에 제시된 Speck32/64(이하 Speck)에 대한 암호분석 기법을 Simon32/64(이하 Simon)에 적용하는 과정을 설명하고 그 결과를 제시한다.

3.1 신경망 구분자

구분자는 특정 차분을 갖는 평문쌍이 암호화된 암호문 데이터와 랜덤 데이터를 식별하는 기능을 한다. 본 절에서는 단일 신경망만을 학습시켜 생성한 구분자와 이를 활용하는 Key averaging 기반 구분자에 대해 설명한다.

이때, 암호문 데이터 (C_0, C_1)은 Simon의 부분 라운드로 암호화된 암호문쌍을 의미하며 랜덤 데이터 (R_0, R_1)은 동일한 길이의 랜덤한 출력값을 의미한다. 기존의 Simon 차분분석 결과 차분특성 확률(differential characteristic probability)이 가장 높다고 알려진 $\Delta P = 0x0000/0020$[16]을 평문 차분으로 설정하여 암호문 데이터를 생성하였다.

3.1.1 기본 구분자

신경망 중 ResNet 구조의 CNN을 기본 구분자로 사용하며, r-라운드 Simon으로 암호화된 암호문 데이터를 랜덤 데이터와 구분하도록 학습된 신경망을 r-라운드 기본 구분자로 정의한다.

3.1.2 Key averaging 기반 구분자

Fig.3.은 r-라운드 기본 구분자를 사용하는 (r+1)-라운드 Key averaging 기반 구분자에 대한 도식화이다. 학습된 r-라운드 기본 구분자를 이용하여 (r+1)-라운드 Key averaging 기반 구분자가 입력 데이터를 식별하는 과정은 다음과 같다.

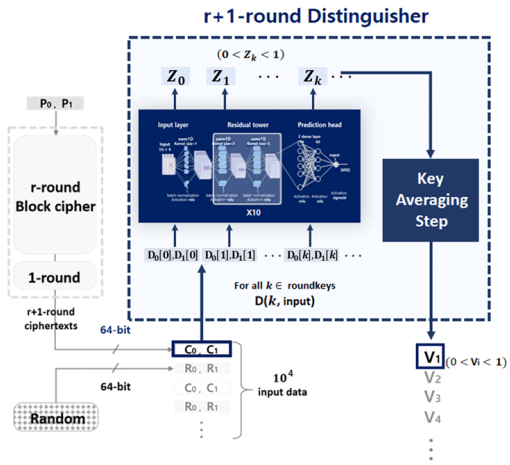

Fig. 3. Process of distinguisher based on Key averaging algorithm

- **복호화 과정**
  하나의 입력 데이터 (C_0, C_1) 또는 (R_0, R_1)을 (r+1) 번째 라운드 키가 될 수 있는 가능한 모든 키 $k \in \{0, 1, 2, ..., 2^{16}-1\}$으로 복호화한다. 복호화된 값을 $(D_0[k], D_1[k])$라고 하며 이 과정에서 총 2^{16}개의 $(D_0[k], D_1[k])$가 출력된다.
- **기본 구분자 예측과정**
  2^{16}개의 $(D_0[k], D_1[k])$가 학습된 기본 구분자의 입력이 된다. 따라서 총 2^{16}개의 신경망의 예측값 z_k가 출력되며, 각 z_k의 범위는 $0 < z_k < 1$이다.
- **Key averaging 단계**
  2^{16}개의 z_k를 Key averaging 단계의 입력하여 출력된 V값이 알고리즘의 최종 출력값이 된다. Key averaging 단계는 2장에서 Key averaging 알고리즘을 설명한 Fig.2. Algorithm1의 4-7단계를 의미한다. 알고리즘의 입력 데이터가 암호문 데이터일 때 나타나는 z_k의 차이를 Key averaging 단계가 극대화하는 것으로 전체 알고리즘의 구분자 성능을 향상시킬 것으로 예상하였다(이는 4.1절에서 더 자세히 설명한다).

3.2 차분분석 결과

A. Gohr의 논문에서 분석한 Speck32/64와 동일한 블록 키 크기를 갖도록 Simon32/64에 대해 실험을 진행하였으며 이에 관한 결과는 Table1. 과 같다. 실험에는 NVIDIA Geforce TITAN X GPU를 사용하였으며 입력 데이터 10^7개에 대한 신경망의 200 에폭(epoch) 학습은 약 8시간 정도가 소요되었다

Table2.에서 CDD는 기존 차분분석 방식을 의미하며 neural network는 기본 구분자를, Key averaging algorithm은 Key averaging 기반 구분자를 의미한다. 측정 결과인 정확도는 구분자가 입력 데이터들 중 올바르게 구분한 데이터의 비율을 나타낸다. Speck 구분자의 경우 [5]에 게재된 정확도 범위 내에서 정확도가 측정됨을 확인하였다. 또한, 해당 방법을 Simon의 구분자로 사용한 결과 Speck의 5, 6, 7-라운드 구분자와 Simon의 7, 8, 9-라운드 구분자가 유사한 정확도로 구분하는 것을 관찰할 수 있었다. Simon의 경우 Speck 암호보다 2라운드 더 높은 9-라운드까지 구분 가능하였다. 따라서 Speck과 유사한 구조를 가지는 Simon에도 딥러닝 기반의 암호분석 기법을 적용 가능함을 확인하였다.

Table 2. Accuracy of distinguisher

		Speck32/64			Simon32/64	
Distinguisher		CDD	Neural network	Key averaging algorithm	Neural network	Key averaging algorithm
Round	5	0.911	0.927	-	-	-
	6	0.758	0.787	0.796	0.999	-
	7	0.591	0.611	0.633	0.938	-
	8	0.512	-	-	0.746	0.816
	9	-	-	-	0.607	0.659
	10	-	-	-	0.500	0.557

IV. 확률분포를 활용한 신경망 구분자의 동작 분석

3장의 실험을 통해 Speck과 Simon 모두 Key averaging 알고리즘을 적용하였을 때 구분자의 성능이 향상되는 것을 확인하였다. 본 장에서는 이러한 성능향상의 원인을 분석한다.

Key averaging 단계는 신경망 구분자의 성능을 높이기 위한 전처리 과정이다. Key averaging 단계의 입력이 되는 데이터들은 특정 분포 특성을 가진다. 이러한 분포 특성은 여러 데이터 셋에 대해 반복 확인하였을 때 재연 가능한 공통적인 특성이며 Key averaging 단계가 이를 극대화 시킨다.

Fig.4.는 Key averaging 단계의 입력이 되는 데이터 셋의 평균적인 분포를 관찰한 것이다. A, B, C, D 그래프는 데이터 셋의 특징에 따라 각각 다른 분포 특성을 가지며, 이러한 특성은 여러 데이터 셋을 출력하여도 유사한 결과가 반복된다. 각 히스토그램의 x축은 z_k값으로 x축의 범위는 z_k의 범위와 동일하게 $0 < x < 1$이다. y축은 x축을 동등하게 분할한 각 구간 $\mathrm{I}_i = \left[\frac{1}{40}i, \frac{1}{40}(i+1)\right), (0 \leq i < 40)$ 내에서 z_k 개수의 평균을 의미한다. 각 구간의 평균과 표준편차를 계산하기 위해 5×10^3개씩 총 3개의 데이터 셋을 관찰하였다.

Fig.4.을 통해 입력 데이터의 종류, 암호 알고리즘의 종류에 따라 일괄적인 분포 특성을 가지며 이에 따라 Key averaging 단계를 통한 전처리 과정이 유의미함을 알 수 있다. 따라서 본 장에서는 Key averaing 단계가 구분자의 성능을 향상시키는 원인을 입력 데이터의 종류에 따라 분석한다. 또한, Key averaing 단계의 내부함수를 가중치 함수(weight function)로 정의하고 이에 대해 분석한다. 다음으로 암호 알고리즘의 라운드 함수 특성에 따라 신경망 예측값의 분포차가 있음을 확인하고 그 원인을 설명한다.

4.1 암호문 데이터의 예측값 분포 특성

Key averaging 단계의 입력이 되는 신경망의 예측값 z_k는 알고리즘의 입력이 랜덤 데이터일 때와 암호문 데이터일 때 분포의 차이를 보인다. 이러한 z_k의 분포 차를 가중치 함수가 극대화함으로써 구분자의 정확도가 향상된다.

4.1.1 암호문 데이터와 랜덤 데이터의 예측값 분포

Fig.5.는 랜덤 데이터 (R_0, R_1) 1개와 6-라운드 Speck으로 암호화된 데이터 (C_0, C_1) 1개에 대한

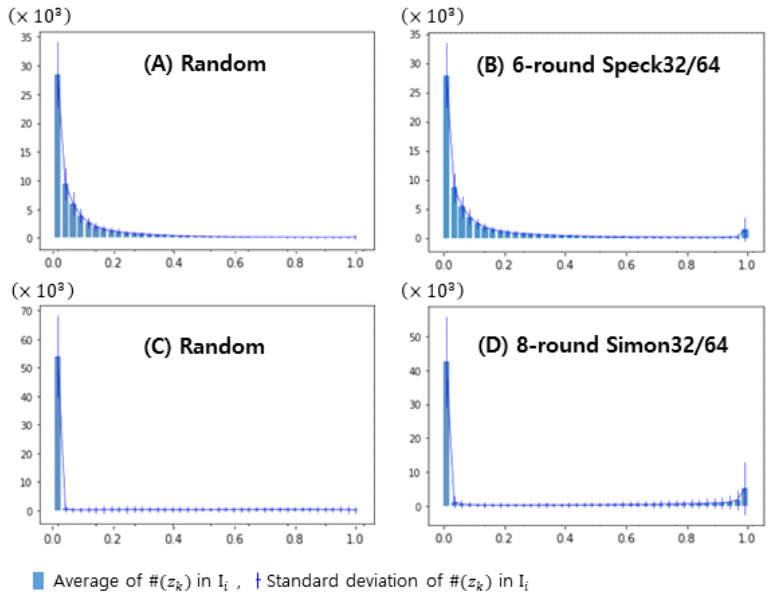

Fig. 4. Average distribution of z_k according to the characteristics of each dataset (A, B are predicted by the Net trained for 5-round Speck32/64 and C, D are predicted by the Net trained for 7-round Simon32/64)

5-라운드 기본 구분자의 예측값 z_k의 분포 그래프를 비교한 것이다. 히스토그램의 x축은 z_k값으로 Fig.4.에서의 정의와 동일하다. x축 구간의 분할은 파이썬(python)의 Seaborn 라이브러리에서 제공하는 함수에 의해 자동으로 결정되도록 하였다.

히스토그램의 y축은 구간 내의 z_k의 개수를 의미한다. Speck는 16-bit 라운드 키를 사용하기 때문에 $k \in \{0,1,2,...,2^{16}-1\}$의 범위에 따라 z_k의 총 개수도 2^{16}개가 된다. 즉, 히스토그램은 하나의 입력 데이터에 대한 2^{16}개의 z_k의 분포를 나타낸다.

입력 데이터가 암호문 데이터일 때 z_k의 분포는 Fig.5.의 우측에 표시된 부분과 같이 랜덤 데이터인 경우와 구분되는 특징을 가진다. z_k는 입력 데이터를 후보 키로 1-라운드 복호화한 후 기본 구분자에 의해 구분된 결괏값이다. (r+1)-라운드 암호문 데이터를 암호화에 사용되었던 올바른 라운드 키 k^*로 복호화할 경우 r-라운드 기본 구분자는 복호화된 결과 $(D_0[k^*], D_1[k^*])$를 r-라운드 암호문 데이터로 구분할 것이다. 따라서 기본 구분자의 구분 결괏값인 z_k는 1에 가까운 값이 출력되어 이러한 분포의 특징이 나타나게 된다. Simon에 대해서도 동일한 방법으로 실험한 결과 Fig.5.와 같이 암호문 데이터에서 1에 가까운 z_k의 값이 출력되었다.

Fig.6.는 랜덤 데이터 (R_0, R_1) 1개와 8-라운드 Simon로 암호화된 데이터 (C_0, C_1) 1개에 대한 7-라운드 기본 구분자의 예측값 z_k의 분포 그래프를 비교한 것이다. Fig.5.와 마찬가지로 히스토그램의

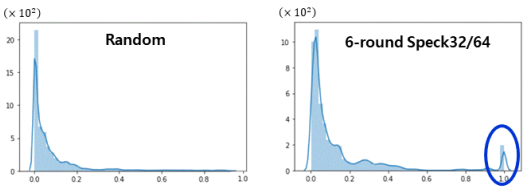

Fig. 5. Neural network prediction value z_k distribution of Speck32/64

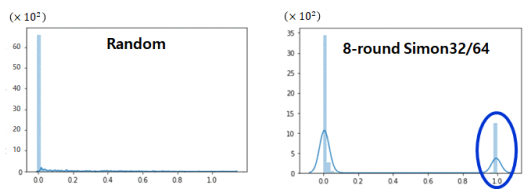

Fig. 6. Neural network prediction value z_k distribution of Simon32/64

x축은 z_k값을, y축은 구간 내의 z_k의 개수를 나타낸다.

4.1.2 가중치 함수를 통한 예측값 분포 극대화

Key averaging 단계에서 가중치 함수는 암호문 데이터의 z_k 분포에서만 나타나는 1에 가까운 z_k값을 증폭시키는 것으로 구분자의 성능을 향상시킨다. 이러한 Key averaging 단계는 각 z_k값에 따라 가중치를 부여하는 함수 f와, 평균을 계산한 후 역함수 f^{-1}를 적용하여 출력값의 범위를 환원시키는 과정으로 나눌 수 있다. 이때, [5]에서는 Fig.7.과 같은 분수함수 $\frac{1}{1-x}$를 가중치를 부여하는 함수로 사용한 것을 확인할 수 있다. Key averaging 단계의 내부에서 가중치를 부여하기 위해 사용되는 가중치 함수 $W(x)$와 이를 사용하는 Key averaging 단계는 Fig.7.과 식 (3)-(4)와 같이 표현할 수 있다.

$$W(x) = \frac{1}{1-x}.$$
$$f(z_k) = W(z_k) \times z_k = v_k. \tag{3}$$

$$V = f^{-1}\left(\frac{1}{2^m}\sum_{k=0}^{2^m-1} f(z_k)\right). \tag{4}$$

위와 같이 가중치 함수를 적용한 Key averaging 단계 결과, 랜덤 데이터의 경우 알고리즘의 출력값이 $V<0.5$일 확률이 높아지고 암호문 데이터의 경우 $V\geq 0.5$일 확률은 높아져 구분의 정확도가 향상된다. Fig.5.와 Fig.6.의 암호문 데이터에 대한 z_k의 분포에서 공통적으로 나타나는 특징을 확인하였다. 이러한 1에 가까운 z_k값은 증가함수인 가중치 함수에 의해 더 큰 가중치가 부여되고, 0에 가까운 값일수록 매우 낮은 가중치가 부여되기 때문에 랜덤 데이터와 암호문 데이터의 z_k의 분포 차이는 더 극대화된다. 결과적으로 z_k의 평균을 계산한 값인 구분자의 결괏값 V의 차이도 더 극대화되며 이로 인해 구분자의 정확도가 향상된다.

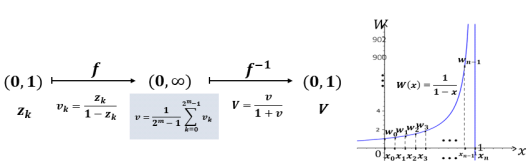

Fig. 7. Key averaging step and example of weight function

4.2 라운드 함수 특성에 따른 예측값 분포

4.1절에서는 구분자의 정확도가 유사한 6-라운드 Speck과 8-라운드 Simon에 대한 신경망의 예측값 z_k의 분포를 관찰하였다. 이때, z_k의 분포는 암호문 데이터를 생성하는 암호 알고리즘의 종류에 따라서도 달라진다. Fig.8.은 암호 알고리즘이 6-라운드 Speck일 경우와 8-라운드 Simon일 경우의 z_k의 분포를 비교한다.

Speck의 경우 z_k가 0과 1 사이의 중간값에도 분포되어있는 것과 다르게 Simon의 경우 z_k의 값이 양극단으로 구분되어 나타났다. 본 절에서는 이러한 분포차의 원인을 확인하기 위해 Speck과 Simon의 1-라운드 복호화 과정 및 이에 따른 차분의 변화를 분석한다.

암호 알고리즘의 r-라운드 암호화 과정을 E_r라고 하자. E_r의 최대 차분확률이 $DP^{E_r}\max = DP^{E_r}(\alpha,\beta)$ 라면, 이러한 r-라운드 차분특성이 신경망의 학습에 반영되어 신경망이 기본 구분자로서의 역할을 할 것으로 가정한다. 즉, 차분확률이 높은 데이터라면 신경망이 암호문 데이터로 구분하여 1에 가까운 z_k값을 출력하는 것으로 예상한다.

Key averaging 알고리즘에서는 이러한 r-라운드 기본 구분자를 활용하여 (r+1)-라운드 구분자를 생성한다. 일반적으로 하나의 (r+1)-라운드 암호문 데이터를 가능한 모든 k로 복호화한 결과 $(D_0[k], D_1[k])$의 차분 $D_0[k]\oplus D_1[k]$은 k에 따라 달라진다. 다시말해 평문의 차분이 $P_0\oplus P_1=\alpha$이고

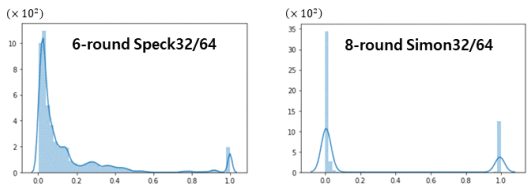

Fig. 8. z_k distribution comparison of Speck32/64 and Simon32/64 (ciphertext data)

r-라운드 암호화 이후의 차분이 $E_r(P_0)\oplus E_r(P_1)=\beta$인 경우라도 (r+1)-라운드 암호문을 1-라운드 복호화한 $D_0[k]\oplus D_1[k]$는 k가 암호화에 사용된 올바른 키일 경우에만 β가 된다.

그러나 Simon의 경우 $D_0[k]\oplus D_1[k]$가 후보 키 k에 무관하게 출력된다. 이에 의해 $E_r(P_0)\oplus E_r(P_1)=\beta$인 경우 1의 확률로 $D_0[k]\oplus D_1[k]=\beta$가 되어 Fig.8.에서 관찰한 것과 같은 분포의 특징을 갖게 된다. 다음 항은 Speck과 Simon의 1-라운드 복호화 과정을 분석하는 것을 통해 후보 키 k와 차분 $D_0[k]\oplus D_1[k]$의 관계를 설명한다.

4.2.1 Speck32/64의 복호화 과정

Speck의 라운드 함수를 통한 암호화 과정은 식 (1)과 같았다. 라운드 함수의 연산 중 비선형 연산인 addition modulo 2^n을 $\mathrm{f}(x,y)=x\boxplus y$로 정의하고 minus modulo 2^n연산을 $\mathrm{f}^-(x,y)=x\boxminus y$로 정의하자. 식 (1)은 다음과 같이 표현 가능하다.

$$\begin{cases} L' = \mathrm{f}((L\gg 8),R)\oplus k \\ R' = (R\ll 3)\oplus \mathrm{f}((L\gg 8),R)\oplus k \end{cases}$$

이때, Speck의 1-라운드 복호화 과정은 다음과 같이 표현 가능하다.

$$\begin{cases} L = \mathrm{f}^-(L'\oplus k,(R'\oplus L')\gg 3)\ll 8 \\ R = (R'\oplus L')\gg 3 \end{cases} \quad (5)$$

식 (5)와 같이 복호화 과정에서 마지막 라운드 키 k는 비선형 연산 f^{-1}의 입력으로 들어가게 된다. 따라서 1-라운드 복호화 이후 $(D_0[k],D_1[k])$의 차분 $D_0[k]\oplus D_1[k]$는 k에 따라 값이 달라진다.

4.2.2 Simon32/64의 복호화 과정

Simon의 경우 라운드 함수를 통한 암호화 과정은 식 (2)와 같았다. Simon의 비선형 연산은 circular shift와 AND 연산이 결합된 $(x\ll 1)\wedge(x\ll 8)$이다. 이러한 연산을 포함하여 함수 $\mathrm{f}(x)=((x\ll 1)\wedge(x\ll 8))\oplus(x\ll 2)$를 정의하자. 식 (2)는 다음과 같이 표현할 수 있다.

$$\begin{cases} L' = (R\oplus \mathrm{f}(L))\oplus k \\ R' = L \end{cases}$$

이때, Simon의 1-라운드 복호화 과정은 다음과 같이 표현 가능하다.

$$\begin{cases} L = R' \\ R = L'\oplus k\oplus \mathrm{f}(R') \end{cases} \quad (6)$$

식 (6)에서 복호화 과정의 마지막 라운드 키 k는 차분을 결정하는 비선형 연산 f와 무관함을 알 수 있다. 따라서 1-라운드 복호화 이후 $(D_0[k],D_1[k])$의 차분 $D_0[k]\oplus D_1[k]$는 k와 무관하게 하나의 값으로 결정된다. 자세한 원인을 Fig.9.와 다음 식들을 통해 설명한다.

- r-라운드 Simon E_r에 대해 입력차분이 α, r-라운드 이후 출력차분이 β인 평문쌍 (P_0,P_1)을 가정하자.

$$P_0\oplus P_1=\alpha,\ E_r(P_0)\oplus E_r(P_1)=\beta.$$

- (P_0,P_1)의 (r+1)-라운드 암호문쌍을 (C_0,C_1)라고 하면 이에 대한 1-라운드 복호화 과정은 다음과 같다.

$$E_r(P_0)=(L_0,R_0),\ E_r(P_1)=(L_1,R_1).$$
$$C_0=(L'_0,R'_0),\ C_1=(L'_1,R'_1).$$

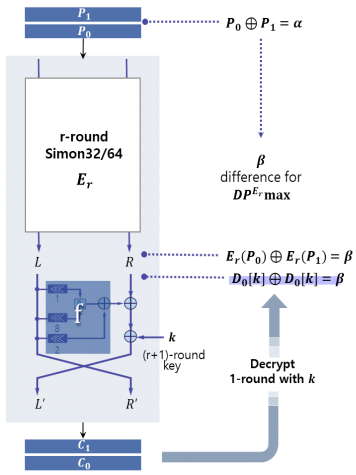

Fig. 9. Differential of 1-round decrypted Simon

(i) **올바른 키 k^*로 복호화하는 경우**

(r+1)-라운드 암호화에 사용된 올바른 키 k^*를 사용하는 경우 (C_0, C_1)는 식 (7)과 같이 r-라운드 암호문으로 복호화된다.

$$(D_0[k^*], D_1[k^*]) = (E_r(P_0), E_r(P_1))$$
$$D_0[k^*] \oplus D_1[k^*] = E_r(P_0) \oplus E_r(P_1) = \beta. \quad (7)$$

또한, 식 (6)에 의해 k^*를 이용한 1-라운드 복호화 과정을 나타낼 수 있으며, 이에 따라 복호문의 차분은 다음과 같이 계산된다.

$$\begin{aligned} & D_0[k^*] \oplus D_1[k^*] \\ &= (R_0' \oplus R_1', (L_0' \oplus k^* \oplus \mathrm{f}(R_0')) \\ &\quad \oplus (L_1' \oplus k^* \oplus \mathrm{f}(R_1'))) \\ &= (R_0' \oplus R_1', (L_0' \oplus L_1') \\ &\quad \oplus (\mathrm{f}(R_0') \oplus \mathrm{f}(R_1'))). \end{aligned} \quad (8)$$

식 (8)을 통해 복호문의 차분은 키와 무관하게 동일한 값으로 복호화된다는 것을 알 수 있다. 식 (7)과 식 (8)은 같은 차분을 의미하므로 다음과 식 (9)가 성립한다.

$$\begin{aligned} & (R_0' \oplus R_1', (L_0' \oplus L_1') \\ & \oplus (\mathrm{f}(R_0') \oplus \mathrm{f}(R_1'))) = \beta. \end{aligned} \quad (9)$$

(ii) **틀린 키 k'로 복호화하는 경우**

임의의 키 $\{0, 1, 2, \ldots, 2^{16}-1\}$ 중 틀린 키 k'를 사용하여 (C_0, C_1)를 복호화하는 경우는 다음과 같다.

$$\begin{aligned} & D_0[k'] \oplus D_1[k'] \\ &= (R_0' \oplus R_1', (L_0' \oplus k' \oplus \mathrm{f}(R_0')) \\ &\quad \oplus (L_1' \oplus k' \oplus \mathrm{f}(R_1'))) \\ &= (R_0' \oplus R_1', (L_0' \oplus L_1') \oplus (\mathrm{f}(R_0') \oplus \mathrm{f}(R_1'))). \end{aligned}$$

위 결과는 올바른 키를 사용하여 복호화하였던 식 (8)과 같다. 따라서 식 (9)에 의해 틀린 키 k'로 복호화한 복호문의 차분도 β가 되는 것을 알 수 있다.

$$D_0[k'] \oplus D_1[k'] = \beta.$$

- 즉, 임의의 키 $k \in \{0, 1, 2, \ldots, 2^{16}-1\}$에 대해 1-라운드 복호화 결과 복호문의 차분은 항상 같은 값으로 결정되며 이는 r-라운드 암호문의 차분과 같다.

이와 같은 이유로 Key averaging 알고리즘의 입력이 (r+1)-라운드 Simon의 암호문 데이터일 경우, 키 k에 상관없이 복호문의 차분은 일정할 것이다. 즉 Simon은 $DP^{E_r}\max = DP^{E_r}(\alpha, \beta)$일 때, r-라운드 이후 차분이 $E_r(P_0) \oplus E_r(P_1) = \beta$라면 복호화 차분도 β가 되어 z_k는 키 k와 무관하게 1에 가까운 값이 출력된다. 그러므로 Speck을 비롯한 일반적인 경우보다 구분자가 차분특성을 통해 암호문 데이터를 구별해 낼 확률이 높아진다.

만약 가정했던 것과 같이 신경망이 차분특성을 학습하여 구분자의 역할을 하는 것이라면 복호화 차분이 키에 무관하게 동일하기 때문에, z_k값 키에 무관하게 유사한 값들이 출력되어야 한다. Fig.10.의 좌측 히스토그램과 같이 높은 차분확률을 갖는 암호문 데이터의 경우 z_k는 모두 1에 가까운 값이 출력된다. 그러나 실제로 대다수의 암호문 데이터에 대해, z_k의 분포는 우측의 그림과 같은 양극단으로 나뉘어 분포된 양상을 보인다. 이 결과를 보아 신경망은 암호문 데이터에 대해 차분특성뿐만 아니라 키 스케줄(key schedule) 과정의 특징과 같은 다른 특성들도 학습하였을 가능성이 있다.

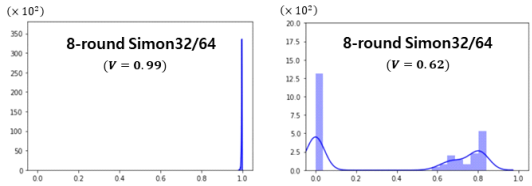

Fig. 10. z_k distribution comparison of Simon according to V value (ciphertext data)

V. 구분자 개선방안

암호 알고리즘의 종류에 따라 암호문 데이터의 신경망 예측값 z_k의 분포는 서로 다른 양상을 보인다. 따라서 랜덤 데이터와 암호문 데이터에 대한 z_k의 분포차를 극대화 시키는 최적의 가중치 함수도 암호 알고리즘의 종류마다 다를 것으로 예상하며 Key averaging 알고리즘을 다른 블록암호에 적용하기 위해 최적의 가중치 함수를 찾는 방법에 관한 연구가 필요할 것으로 보인다.

5.1 가중치 함수 선정

Key averaging 알고리즘의 입력이 각각 랜덤 데이터 (R_0,R_1)일 때와 암호문 데이터(C_0,C_1)일 때, 신경망 Net의 예측값 z_k의 집합을 다음과 같이 표현할 수 있다(단, $k\in\{0,1,\cdots,2^{16}-1\}$).

$$Z_{Rand}=\{z_k\,|\,z_k=\mathrm{Net}(\mathrm{Dec1R}((R_0,R_1),k))\}.$$
$$Z_{Cipher}=\{z_k\,|\,z_k=\mathrm{Net}(\mathrm{Dec1R}((C_0,C_1),k))\}.$$

Fig.11.은 Speck과 Simon의 Z_{Rand}와 Z_{Cipher}의 분포를 비교하고, 값의 구간을 사분위로 나누어 각 구간에 속하는 z_k의 수를 관찰한 결과를 나타낸다. 이때, 사분위로 나눈 그래프의 각 구간의 수치는 10^4개의 데이터에 대한 평균값이다. Z_{Rand}가 대부분 출력되는 좌측 구간을 (구간1)이라 하고, 거의 출력되지 않는 우측 구간을 (구간2)라 하자. 효율적인 가중치 함수를 선정하기 위해서는 (구간1)에는 0에 가까운 가중치를 두고 (구간2)에는 1에 가까운 높은 가중치를 부여하여, (구간2)의 Z_{Cipher}의 z_k값을 극대화하도록 해야한다. 따라서 효율적인 가중치 함수의 선정에는 다음과 같은 요소를 고려해야 한다.

- Z_{Rand}의 분포 관측을 통한 (구간1)과 (구간2)의 분할
- Z_{Cipher}의 z_k값이 출력되는 (구간2)에서의 가중치 상승 정도

이때 (구간1), (구간2)의 범위는 블록암호 알고리즘에 따라 달라지며 Simon의 경우 Speck에서 사용한 분수함수가 최적의 가중치 함수가 아닐 것으로 예상하였다.

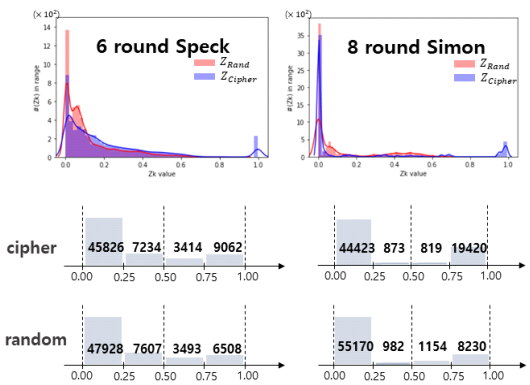

Fig. 11. Comparison of Z_{Rand}, Z_{Cipher} distribution between Speck and Simon

본 연구에서는 분수함수를 대체할 가중치 함수를 Tangent, Relu, Step 함수로 선정하였으며 실험적으로 각 가중치 함수의 적합성을 확인하였다.

① Tangent: $W(x)=2\tan\left(\frac{\pi}{2}x\right)$

② Relu: $W(x)=\mathrm{relu}(x,a,b)=\max(0,b(x-a))$

③ Step: $W(x)=\mathrm{step}(x,a,b)=\begin{cases}a, x>b\\0, x\le b\end{cases}$

5.2 가중치 함수 대체실험

Table 3.은 Speck을 대상으로 한 가중치 함수 대체실험 결과로 각 구분자의 정확도를 기재한 것이다. 각 정확도의 측정은 3.2절의 Table 2.의 실험과 동일하게 진행하였다. 이때, 각 가중치 함수 W에 대해 (구간2)에서의 가중치 상승 정도를 파악하기 위해 W의 함수값과 W'를 함께 기재하였다. 본 실험에서 사용한 Tangent, Relu, Step 함수 중 기존의 Key averaging 알고리즘에 사용된 분수함수 $\frac{1}{1-x}$보다 높은 성능을 보이는 함수는 없었다. 이는 가중치 함수 중 $\frac{1}{1-x}$가 암호문 데이터에서만 나타나는 분포 특성을 가장 효율적으로 극대화하였기 때문이다. 암호문 데이터에서만 나타나는 1에 가까운 z_k값은 가중치 함수의 기울기 등의 특성에 따라 증폭되는 정도가 달라진다. $\frac{1}{1-x}$는 Speck의 암호문 데이터에서 나타나는 분포 특성을 극대화하는데 실험에 사용된 가중치 함수 중 가장 적합한 것으로 보인

Table 3. Accuracy of neural distinguisher for Speck32/64 according to the weight functions

Distinguisher		Neural network	$\frac{1}{1-x}$	Tangent	Relu	Step
Round	5	0.927	-	-	-	-
	6	0.787	**0.796**	0.784	0.777	0.557
	7	0.611	**0.633**	0.603	0.621	0.534
$W(0.9)$		-	10.0	12.5	10.0	100.0
$W'(0.9)$		-	100.0	127.1	50.0	0.0

Table 4. Accuracy of neural distinguisher for Simon32/64 according to the weight functions

Distinguisher		Neural network	$\frac{1}{1-x}$	Tangent	Relu	Step
Round	7	0.938	-	-	-	-
	8	0.746	0.816	**0.822**	0.786	0.818
	9	0.607	0.659	**0.654**	0.634	0.518
$W(0.9)$		-	10.0	12.5	10.0	100.0
$W'(0.9)$		-	100.0	127.1	50.0	0.0

다. 따라서 각 함수의 파라미터로 설정하였던 값을 조절할 경우 구분 정확도를 더 높일 수 있을 것으로 예상되지만, 현재까지의 실험 결과로 보아 분수함수 $\frac{1}{1-x}$가 Speck의 암호분석에 적절한 가중치 함수로 판단된다.

Table 4.는 마찬가지로 Simon을 대상으로 한 가중치 함수 대체실험 결과이다. 본 실험에서는 Tangent 함수가 암호문 데이터와 랜덤 데이터 구분에 가장 좋은 성능을 보였다. 이는 분수함수를 사용하였을 때 최적의 성능을 보였던 Speck에서의 결과와는 차이가 있는 것으로 Fig.8.과 같은 신경망 예측값 z_k의 분포 차이에 의해 최적의 가중치 함수가 달라지는 것으로 보인다. 그러나 여전히 분수함수나 Tangent 함수가 최적화된 가중치 함수라고 말할 수 없으며, 각 함수의 파라미터로 설정하였던 값에 따라 더 높은 정확도를 보이는 함수가 존재할 수 있다.

VI. 결 론

본 논문에서는 대칭키 암호 시스템의 블록암호 안전성 분석에 딥러닝 기술을 활용한 동향을 파악하고 그 중 A. Gohr가 제시한 암호분석 방법의 동작 원리를 분석하여 개선 방향을 제시하였다. 또한, 경량 블록암호 Simon에 이를 적용한 결과를 제시하였다.

우선 Gohr가 제시한 암호분석 방식에서 신경망 예측값의 확률분포가 입력 데이터에 따라 상이함을 관찰하였다. 첫째로 특정 암호 알고리즘에 대해 구분자로 학습된 신경망을 구분자로 사용할 때, 입력 데이터가 랜덤 데이터인지 암호문 데이터인지에 따라 예측값의 확률분포는 달라진다. 암호문 데이터의 예측값 확률분포에서만 나타나는 특성이 있으며, 그 분포 특성을 Key averaging 알고리즘 내부의 가중치 함수가 극대화하여 구분자의 성능이 향상됨을 설명하였다. 두 번째로 암호 알고리즘의 종류에 따라서도 확률분포의 양상이 달라짐을 관찰하고 그 원인을 설명하였다. Speck과 Simon 각각에 대해 구분자로 학습된 신경망을 사용하였을 때, 암호문 데이터에 대한 신경망 예측값의 확률분포는 서로 다른 양상을 보인다. 이러한 차이는 Speck과 Simon의 라운드 함수 특성에 의한 것으로, 두 암호 알고리즘의 복호화 과정을 분석하는 것을 통해 이를 설명하였다.

또한, 암호 알고리즘에 따른 확률분포의 차이에 의해 최적의 가중치 함수 또한 차이가 있을 것으로 예상하고 이를 실험적으로 확인하였다. 즉, 각 암호 알고리즘에 적합한 가중치 함수를 선정하는 것을 통해 암호분석의 핵심기술인 구분자의 성능을 향상 시킬 수 있을 것으로 보인다. 따라서 암호 알고리즘마다 신경망 예측값의 확률분포에 따라 가중치 함수를 선정하는 방법론을 좀 더 체계적으로 도출할 필요가 있어 보이며 본 연구가 이에 관한 기반연구로 활용될 것으로 기대한다.

References

[1] M.M. Alani, "Neuro-cryptanalysis of DES and triple-DES," In International Conference on Neural Information Processing, pp. 637-646, Nov. 2012.

[2] R. Focardi, and F.L. Luccio, "Neural cryptanalysis of classical ciphers," Italian Conference on Theoretical Computer Science, pp.104-115, Sep. 2018.

[3] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural networks, vol. 2, no. 5, pp. 359-366, 1989.

[4] S. Kwon, H. Yim, J.S. Kang, Y. Yeom, "Revisiting cryptnalysis of neural plaintext recovery attack of DES," KICS, 46(7), pp. 1109-1119, Jul. 2021.

[5] A. Gohr, "Improving attacks on round-reduced Speck32/64 using deep learning," In Annual International

Cryptology Conference, pp. 150-179, Aug. 2019.
〔6〕 S. Baek, and K. Kim. "Recent advances of neural attacks against block ciphers," 2020 Symposium on Cryptography and Information Security, Jan. 2020.
〔7〕 Z. Hou, J. Ren, and S. Chen, "Cryptanalysis of round-reduced SIMON32 based on deep learning," IACR Cryptol. ePrint Arch, Mar. 2021.S
〔8〕 J. So, "Deep learning-based cryptanalysis of lightweight block ciphers," Security and Communication Networks, vol. 2020, pp. 1-11, Jul. 2020.
〔9〕 A. Jain, V. Kohli, G. Mishra, "Deep learning based differential distinguisher for lightweight block ciphers," arXiv preprint arXiv:2112.05061, Dec. 2021.
〔10〕 M. Wang, "Differential cryptanalysis of reduced-round PRESENT," International Conference on Cryptology in Africa. Springer, Berlin, Heidelberg, vol. 5023, pp. 40-49, Jun. 2008.
〔11〕 A. Baksi, "Machine learning-assisted differential distinguishers for lightweight ciphers," Classical and Physical Security of Symmetric Key Cryptographic Algorithms. Springer, Singapore, pp. 141-162, Jan. 2022.
〔12〕 J. Lu, et al. "Improved neural distinguishers with (related-key) differentials: applications in SIMON and SIMECK," arXiv preprint arXiv:2201.03767, Jan. 2022.
〔13〕 R. Beaulieu, et al. "The SIMON and SPECK lightweight block ciphers," Proceedings of the 52nd Annual Design Automation Conference, pp. 1-6, Jun. 2015.
〔14〕 E. Biham, and A, Shamir, "Differential cryptanalysis of DES-like cryptosystems," Journal of CRYPTOLOGY, vol. 4, no. 1, pp. 3-72, Jan. 1991.
〔15〕 GitHub, "agohr/deep_speck," https://github.com/agohr/ deep_speck, last accessed Mar. 16. 2022.
〔16〕 F. Abed, E. List, S. Lucks, and J. Wenzel, "Differential cryptanalysis of round-reduced Simon and Speck," In International Workshop on Fast Software Encryption, pp. 525-545, Mar. 2014.

〈저 자 소 개〉

성 효 은 (Hyoeun Seong) 정회원
2019년 8월: 국민대학교 수학과 학사
2021년 8월: 국민대학교 일반대학원 금융정보보안학과 석사
〈관심분야〉 암호구현 및 분석, 병렬 프로그래밍, AI 보안

유 현 도 (Hyeondo Yoo) 정회원
2021년 2월: 국민대학교 정보보안암호수학과 학사
2021년 3월~현재: 국민대학교 일반대학원 금융정보보안학과 석사과정
〈관심분야〉 암호구현, 난수성 분석 및 평가

염 용 진 (Yongjin Yeom) 종신회원
1991년 2월: 서울대학교 수학과 졸업
1994년 2월: 서울대학교 수학과 석사
1999년 2월: 서울대학교 수학과 박사
2000년 4월~2012년 2월: ETRI 부설연구소 책임연구원/팀장
2012년 3월~현재: 국민대학교 과학기술대학 정보보안암호수학과 정교수
2013년~현재: 국민대학교 BK21+ 안전한 초연결사회를 위한 문제해결형 정보보안 교육연구단 교수
〈관심분야〉 암호구현 및 분석, 보안시스템 평가

강 주 성 (Ju-Sung Kang) 종신회원
1989년 2월: 고려대학교 수학과 졸업
1991년 2월: 고려대학교 일반대학원 수학과 석사
1996년 2월: 고려대학교 일반대학원 수학과 박사
1997년~2004년: 한국전자통신연구원 선임연구원/팀장
2004년 3월~현재: 국민대학교 과학기술대학 정보보안암호수학과 정교수
2013년~현재: 국민대학교 BK21+ 안전한 초연결사회를 위한 문제해결형 정보보안 교육연구단 교수
〈관심분야〉 암호이론, 정보보안 프로토콜, 안전성 분석 및 평가